VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2017 Bc. Miloš Dolejší

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

ŘÍZENÍ BAREVNÉHO GRAFICKÉHO LED DISPLEJE POMOCÍ FPGA
FPGA CONTROLLER FOR LED VIDEO DISPLAY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. Miloš Dolejší

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Pavel Hanák, Ph.D.

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Miloš Dolejší ***ID:*** 155151
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Řízení barevného grafického LED displeje pomocí FPGA

POKYNY PRO VYPRACOVÁNÍ:

V jazyce VHDL vytvořte modul, který bude provádět tyto funkce:
- Čtení obrazových dat z paralelního rozhraní a jejich ukládání do paměti (frame bufferu).
- Čtení dat z paměti, jejich převod do sériového formátu a zasílání do budičů LED typu SM16126, MBI5024, TPIC6C595 nebo podobných.
- Řízení jasu každého sub-pixelu pomocí PWM.
- Řízení globálního jasu displeje.
- Gama korekci dle předem definovaných tabulek.

Kód koncipujte tak, aby bylo možné snadno změnit rozlišení, barevnou hloubku a jiné parametry displeje. Při návrhu počítejte s maximálním rozlišením 320x240 px a barevnou hloubkou 4 bity na barvu. Funkci modulu ověřte na výrobním prototypu LED displeje, který dodá vedoucí práce.

DOPORUČENÁ LITERATURA:

[1] AKINS, G., RGB LED Panel Driver Tutorial. Dostupné online na http://bikerglen.com/projects/lighting/led-panel-1up/

[2] Adafruit 32x16 and 32x32 RGB LED Matrix Tutorial. Dostupné online na https://learn.adafruit.com/32x16-32x32-rgb-led-matrix/overview

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 24.5.2017

Vedoucí práce: Ing. Pavel Hanák, Ph.D.
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato diplomová práce se věnuje řízení barevného grafického LED displeje pomocí FPGA. V teoretické části byly popsány vlastnosti použitého FPGA, zdroj dat a princip řízení RGB LED displeje. V první polovině teoretické části byly popsány vlastnosti použitého FPGA, zdroj dat a princip řízení RGB LED displeje. Druhá polovina se zabývá gamma korekcí, implementací pulsně šířkové modulace a binární kódové modulace tak, aby bylo možné řídit globální jas displeje a barevnou hloubku každého sub-pixelu. Praktická část se věnuje návrhu a realizaci tohoto modulu v jazyce VHDL, zápisu obrazových dat do paměti z procesoru Blackfin přes rozhraní PPI, následnému čtení dat z paměti, jejich převodu do sériového formátu a zasílání do řadiče LED. Modul byl realizován na vývojové desce Digilent Atlys s FPGA Spartan-6 a odzkoušen na panelu 32x20 pro firmu Ing. Ivo Herman, CSc.

KLÍČOVÁ SLOVA

FPGA, LED displej, VHDL, BRAM

ABSTRACT

This thesis deals with controlling a color graphic LED display using an FPGA. The first half of the theoretical part of this paper describes the properties of the used FPGA, the data source and a principle of controlling an RGB LED display. The second half describes an implementation of pulse width modulation and binary code modulation which enables the control of brightness of the display and of color depth of every sub-pixel. The practical part on the other hand describes the designing and the implementation of this module in the VHDL language. Then it explains the transfer of image data from Blackfin processor to the memory via PPI interface, the subsequent process of reading data from the memory, conversion of the data to a serial format and finally it describes the process of sending the data to the LED controller. The module was realized on the Digilent Atlys development board equipped with the Spartan-6 FPGA and was tested on a 32x20 light panel for the firm Ing. Ivo Herman, CSc.

KEYWORDS

FPGA, LED display, VHDL, BRAM

DOLEJŠÍ, Miloš *Řízení barevného grafického LED displeje pomocí FPGA*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, Rok. 56 s. Vedoucí práce byl Ing. Pavel Hanák, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Řízení barevného grafického LED displeje pomocí FPGA“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Pavlovi Hanákovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych rád touto cestou poděkoval panu Ing. Milanovi Vajdíkovi za poskytnutí cenných informací ohledně testování modulu a firmě Ing. Ivo Herman, CSc. celkovou výbornou spolupráci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Cílem diplomové práce bude vytvořit VHDL modul, který bude řídit barevný RGB displej pomocí FPGA tak, aby bylo možné regulovat jas a barevnou hloubku každého sub-pixelu. Načtená obrazová data z paralelního rozhraní budou ukládána do paměti, zpracována a následně sériově zasílána do řadiče LED displeje. Celý systém bude navržen tak, aby bylo možné jednoduše měnit jednotlivé parametry.

Teoretická část bude věnována rozborem principů jednotlivých komponentů. Obrazová data bude generovat procesor Analog Devices Blackfin, které bude využívat PPI rozhraní pro přenos dat. Obrazová data budou ukládána do dvojité snímkové blokové RAM paměti. Následně budou data čtena a na displeji zobrazena. Celý systém bude řízen programovatelným obvodem FPGA od Xillinx, konkrétně Spartan 6. Ke zvýšení bitové hloubky bude potřeba použít pulsně šířkovou modulaci nebo binární kódovou modulaci, která bude využita i ke změně jasu displeje.

Praktická část bude věnována způsoby návrhu řešení FPGA modulu. Budou zde popsány simulace komunikace mezi zdrojem dat a FGPA a také komunikace mezi FPGA a displejem. Kód bude naprogramován jazykem VHDL, v prostředí Xilinx ISE. Toto prostředí bude také využito na syntézu, implementaci designu, vytvoření spustitelného kódu a v neposlední řadě i na simulaci.

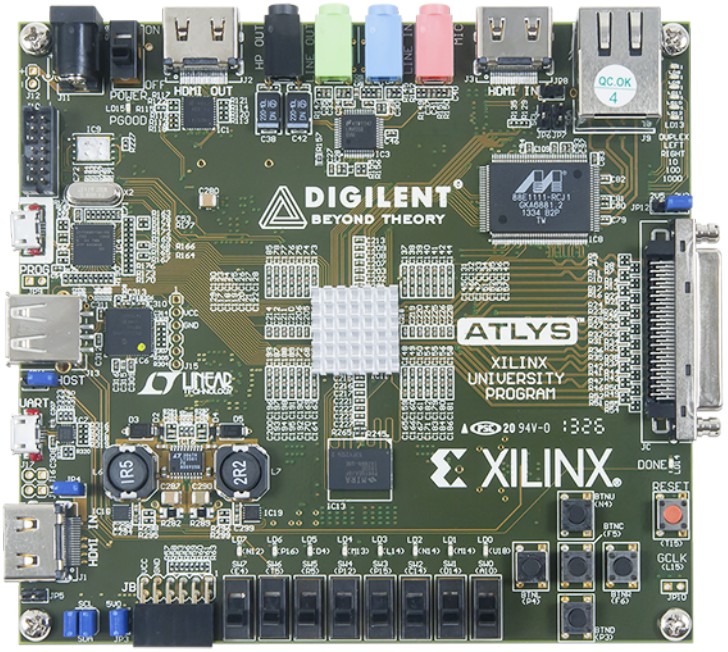
V rámci semestrálního projektu bude ověřena funkčnost na LED displeji Adafruit 32x32. V diplomové části bude naprogramovaný VHDL modul využit pro ovládání na dodaném LED panelu 32x20, ve spolupráci s firmou Ing. Ivo Herman, CSc.

1 POPIS JEDNOTLIVÝCH KOMPONENTŮ

V této části budou popsány komponenty, které byly použity v semestrální a následně v diplomové práci.

1.1 Vývojová deska Digilent Atlys

Vývojová deska od Digilent Atlys(viz obr 1.1) s kombinací FPGA Spartan-6 a jeho periferiemi je vhodná pro tvorbu současných digitálních systémů. Vývojová deska disponuje těmito periferiemi: vysokorychlostní HDMI porty, USB porty, audio porty nebo gigabitový ethernet. Na desce je umístěn FPGA čip XC6SLX45, jehož základní parametry jsou popsané v tabulce 1.1[2].



Obr. 1.1: Vývojová deska použitá v semestrální práci[2].

1.1.1 Vlastnosti FPGA

- **Programovatelná logická hradla**: Tvořena z velkého množství konfigurovatelných bloků obecné logiky
- **Opětovné přeprogramování**: Neomezuje se počtem nahrání programu. Vhodné pro testování programu na reálném zařízení.

Tab. 1.1: Přehled informací několika typů FPGA Spartan-6 [13].

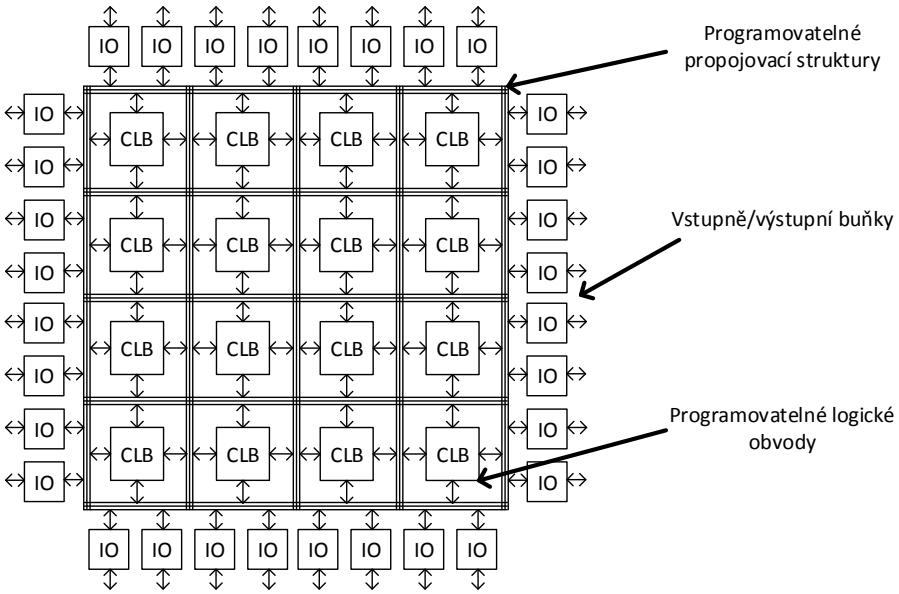
Zařízení	Logické buňky	Konfigurovatelné logické bloky			Bloková RAM paměť		Maximum uživatelských I/O
		Slicy	Rregistry	Maximum Distribuované RAM	16kb	Maximum RAM kb	
XC6SLX25	24,051	3,758	30,064	229	52	936	266
XC6SLX45	43,661	6,822	54,576	401	116	2,088	358

- **Rychlá reakční doba**: FPGA má reakční odezvu řádově ns, na rozdíl od mikroprocesorů, který mají odezvu řádově ms. FPGA nemají instrukční sadu a jsou přímo závislé na frekvenci tiku hodin. Signál se šíří pouze přes logické obvody.
- **Společný čip pro více projektů najednou**: Uživatel si vybere pouze části obvodu, které by splňovaly předem definovanou funkci - hlavní rozdíl proti obvodů ASIC.
- **Bloková RAM**: Speciální paměťové bloky 9kb a 18kb s duálním přístupem (Spartan-6).
- **Distribuovaná RAM**: Může takto fungovat s použitím LUT tabulek a logických obvodů.
- **Jazyky HDL**: Nejčastěji programovatelné jazyky HDL jsou VHDL nebo Verilog[7].

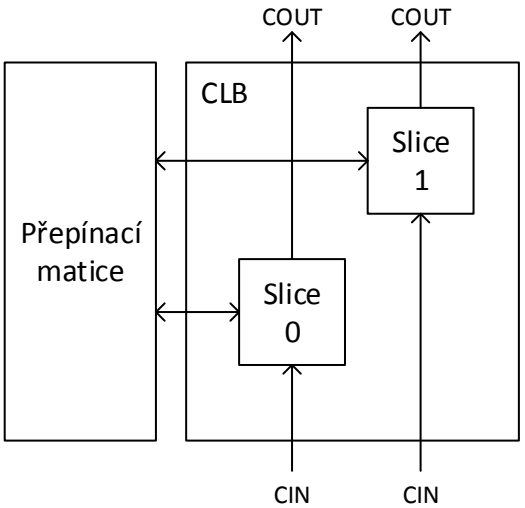
1.1.2 Základní architektura FPGA

Architektura FPGA (viz obr. 1.2) se skládá z *Programovatelných logických buněk*, na obrázku označovaných jako CLB, *Programovatelné propojovacích struktur* a *Vstupně/výstupných buněk*, označkované jako IO.

- **Programovatelné logické buňky**: Jsou tvořeny sekvenční a kombinační částí. Budeme-li mluvit o Spartan 6 CLB obsahuje dva logické bloky (viz obr 1.3), označované jako SLICE. Každý SLICE je složen z šestivstupé LUT tabulky a klopných obvodů, respektivě čtyřmi LUT a osmi klopných obvodů. LUT obvykle za pomocí mintermů realizuje kombinační funkci. Může se použít i jako posuvný registr nebo distribuovanou paměť RAM. Avšak v diplomové práci bude použita bloková RAM. Klopný obvod tvoří sekvenční část bloku a často bývá typu D, který je řízen hranou(Označovány jako FLIP-FLOP) nebo úrovní (označovány jako LATCH), viz dále.
- **Programovatelná logická struktura**: Propojují jednotlivé funkční bloky. Jsou rozděleny do tří skupin: globální, lokální a speciální.
- **Vstupně/výstupní buňky**: Starají se o komunikaci jádra s vnějším prostředím. Díky nim FPGA podporuje například více logických nebo diferenčních



Obr. 1.2: Blokové schéma zapojení FPGA [9].

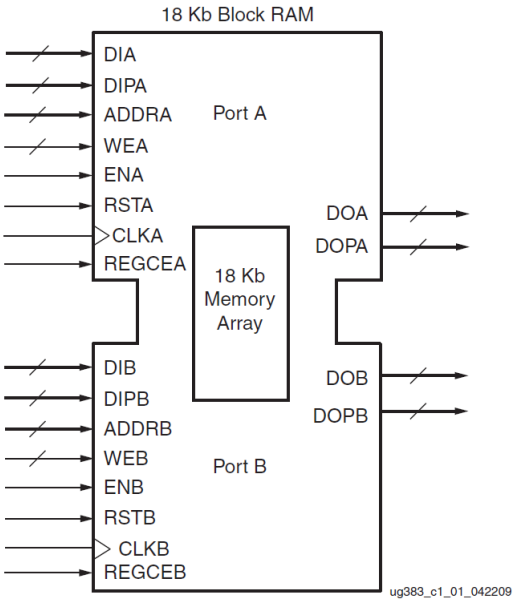


Obr. 1.3: Schéma CLB bloku [11].

standardů.

1.1.3 Bloková RAM paměť

- Obdobně Složena z LUT tabulek.
- Speciální RAM paměť pro rozsáhlejší paměťové struktury (viz obr 1.4).
- Tvořena 18kb bloky, která jsou konfigurovaná jako dvě samostatné 9kb RAM paměti (Spartan-6).
- Obsahuje registry pro zvýšení výkonu.
- Celkový počet BRAM pamětí je závislá na velkosti čipu.
- Vhodná pro rychlé CACHE paměti, zásobníky, fronty nebo ROM paměti.
- Fungují na pracovní rychlosti hodin.
- Tři způsoby čtení a zápisu dat - READ_FIRST, WRITE_FIRST, NO_CHANGE[6].



Obr. 1.4: Bloková RAM paměť - dual port [12].

1.1.4 Druhy logických obvodů

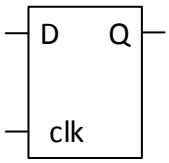
Jak již bylo řečeno, programovatelné logické obvody se skládají z kombinačních nebo sekvenčních logických obvodů.

Kombinační logické obvody

Nemají vnitřní paměť. Jedná se o propojení hradel typu NOT, AND, OR atd. Mohou být popsány za pomocí Karnaughovy mapy, ale i také pravdivostní tabulkou

Sekvenční obvody

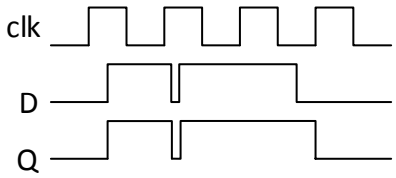
Hlavní prvek sekvenčních obvodů je klopný obvod. Obvykle se v FPGA setkáme s klopnými obvody typu D. Klopné obvody mají paměť a mohou být řízeny staticky a dynamicky. Kvůli dodržení časových parametrů nejčastěji však bývají řízeny dynamicky.



Obr. 1.5: Schématická značka staticky řízeného klopného obvodu D.

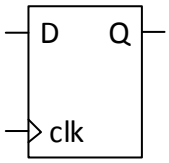
Staticky řízené klopné obvody(obr. 1.5): Též nazývaný jako Latch. Staticky řízené klopné obvody jsou řízené úrovní, které často způsobují zákmity a destabilizují systém.

Z časových průběhů na obr. 1.6 je vidět, že výstupní hodnota klopného obvodu se mění při horní logické úrovni. Zákmit, který se objevil na vstupu, se okamžitě projeví i na výstupu. Tento zákmit není většinou chtěný a proto je vhodnější použít dynamicky řízené klopné obvody [7].

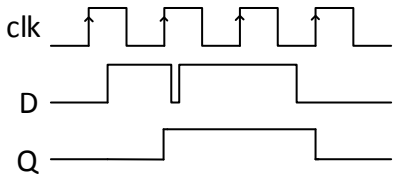


Obr. 1.6: Časové průběhy staticky řízeného klopného obvodu D.

Dynamicky řízené klopné obvody (obr. 1.7): Jsou nazývány FLIP-FLOP a řízeny hranou. Na obr. 1.8 jsou znázorněny časové průběhy dynamicky řízeného kopného obvodu, ze kterých je patrné, že zákmit, který byl na vstupu, se neprojeví na výstupu, protože výstup je řízen pouze na náběžnou, nebo souběžnou hranou hodinového signálu [7].



Obr. 1.7: Schématická značka dynamicky řízeného klopného obvodu D.



Obr. 1.8: Časové průběhy dynamicky řízeného klopného obvodu D..

1.1.5 Metody zvyšování pracovní frekvence

Pro správnou funkcionalitu kopných obvodů, které jsou řízeny náběžnou hranou hodinového signálu, musí být dodrženy požadavky na časování jeho vstupních signálů. Hodinový signál závisí na těchto požadavcích:

- Minimální doba trvání logické jedničky.
- Minimální doba trvání logické nuly.
- Minimální perioda (Maximální frekvence).
- Maximální doba náběžné a sestupné hrany.

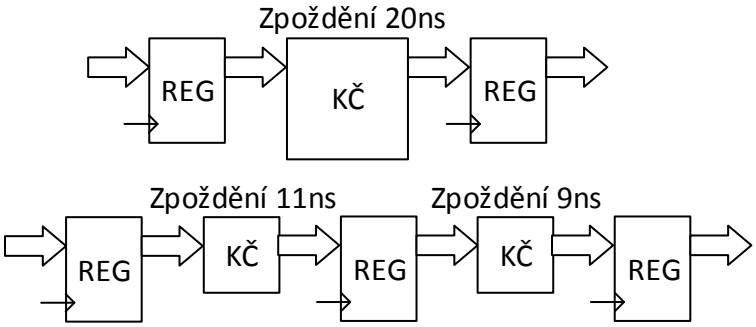
Mezní frekvence je podmíněna největším zpoždění mezi dvěma klopnými obvody (kombinační částí a propojovací strukturou).

Pipelining

První z technik zvyšování pracovní frekvence. Jeho princip je založen na rozdělení velkých kombinačních částí na jednodušší (viz obr. 1.9).

Velké kombinační části vykazují velké zpoždění. Rozdělení na menší kombinační části, společně přidáním mezi těchto částí registry, způsobí snížení zpoždění, které je dáno největším zpožděním kombinační části.

Nevýhodou pipeliningu je využití více klopných obvodů, též také zpoždění výstupního signálu o násobek hodinových taktů vlivem přidání dalších stupňů klopných obvodů.

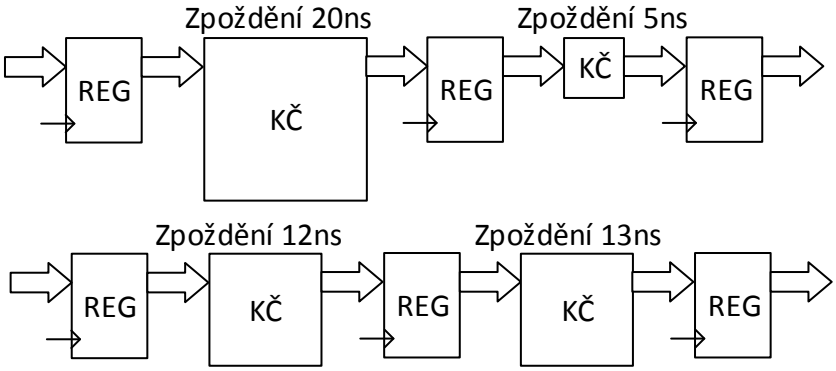


Obr. 1.9: Blokové schéma příkladu pipeliningu.

Register retiming (balancing)

Druhou technikou je balancování registrů (register balancing). Princip je založen na přesunutí části z velké kombinační části (obr. 1.10), která vykazuje velké zpoždění, do jiné kombinační části, jehož zpoždění je malé. Tím se vyrovná zpoždění a docílí se zvýšení pracovního kmitočtu [7].

Tuto techniku je možné aplikovat automaticky. Povolením v nastavením navrhovaného systému.



Obr. 1.10: Blokové schéma příkladu register balancingu.

Register replication

Třetí a zároveň poslední technikou je replikace registrů (register replication), která umožňuje nahradit jediný registr instancí více registrů v částech, kde jsou cílové

obvody. Tato technika se využívá v případě, kdy je v systému signál přiváděný k velkému množství obvodů, které jsou umístěny na velké ploše čipu. Replikací registrů vede ke snížení zátěže a především ke zvýšení pracovní frekvence [7].

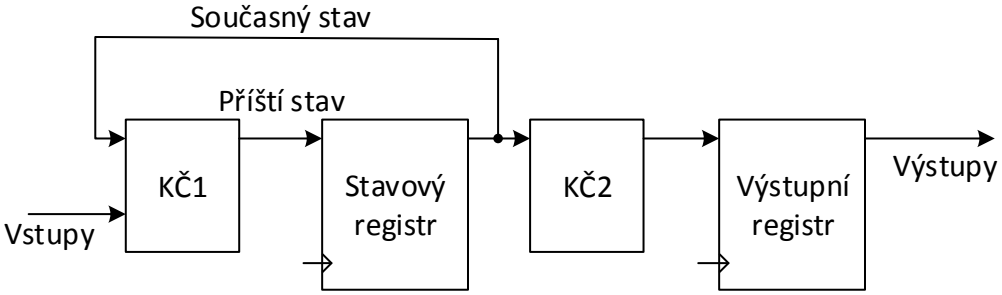
1.1.6 Stavové automaty

Jedná se o konečné stavové automaty. Jsou chápány jako synchronní sekvenční číslicové systémy, které se mohou se nacházet pouze v jednom definovaném stavu.

Stavový automat přechází mezi jednotlivými stavy, dle jeho vstupů a podle stavu generuje výstupy. Skládá se ze stavového registru a kombinačních částí KČ1 a KČ2. KČ1 je kombinační část pro stanovení následujícího stavu a výstupní kombinační část KČ2, která podle současného stavu rozhoduje stav výstupů, jehož výstupy mohou být přivedeny k registrovému výstupu nebo s kombinačním výstupem, které mají na výstupu pouze kombinační část [7].

Moorův stavový automat

Stav automatu je závislý pouze na současném stavu (viz obr. 1.11).



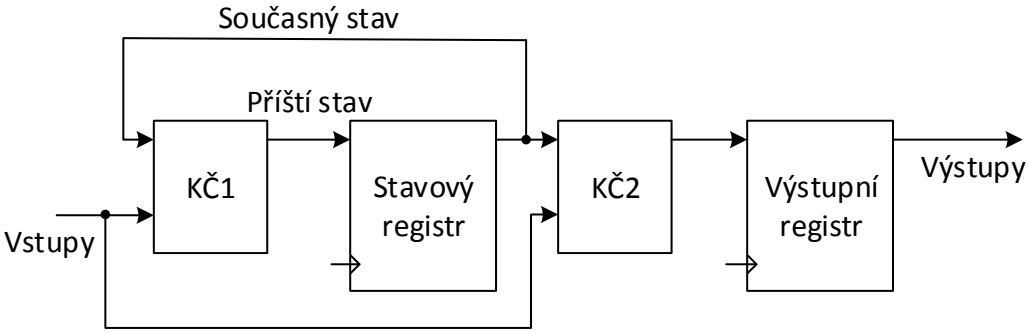
Obr. 1.11: Blokové schéma moorova stavového automatu.

Mealyho stavový automat

U Mealyho stavového automatu stav automatu je závislý jednak na aktuálním stavu, ale také na stavu vstupů (obr. 1.12).

1.2 Zdroj obrazových dat

Zdrojem obrazových dat budou procesory Analog Devices Blackfin. Jedná se 16 nebo 32 bitového procesoru, speciálně navrženého tak, aby splňoval výpočetní výkon



Obr. 1.12: Blokové schéma mealyho stavového automatu.

dnešních audio, video a komunikačních aplikací. Tento procesor bude používat PPI rozhraní ke komunikovat s FPGA [10].

1.2.1 PPI rozhraní

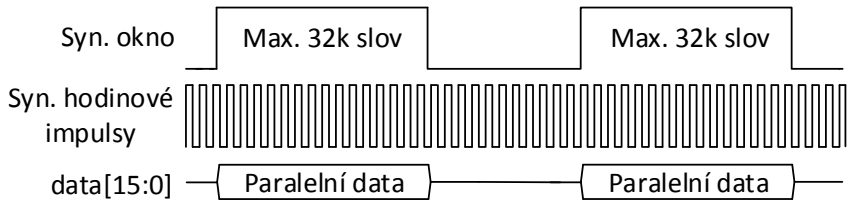
PPI rozhraní je realizováno jako half-duplexní[1] spojení, které má paralelní datový port s maximálně šestnáctibitovou šířkou. Dále má vyhrazený pin pro synchronizační hodiny a až tři synchronizační piny. PPI rozhraní ke správnému fungování požaduje externí generování hodin, jehož maximální dosažitelná frekvence k přenesení jednoho okna dat je 50MHz. Maximální dosažitelná přenosová rychlost je použitím 20-25 MHz generátoru hodinových impulsů [10].

Na obr. 1.13 je vyobrazena komunikace tohoto rozhraní. Jak již bylo zmíněno, jedná se obousměrnou komunikaci, ale obě strany nemohou vysílat současně. Z toho plyne, že jeden z komunikujících vysílá požadavky, a též celou komunikaci řídí. Druhá strana požadavky přijímá a generuje odpovědi. V reálu to vypadá tak, že vysílací strana generuje hodinové impulsy, současně vysílá synchronizační okno, na které přijímací strana v tomto okně vysílá data. Data mohou být dlouhá maximálně 32 tisíc slov. Tyto datová slova jsou synchronizovaná právě s hodinovými impulsy řídící strany.

1.3 LED displej Adafruit

První z obrazových panelů, na kterém se budou zobrazovat obrazová data, je displej Adafruit. Displej má rozlišení 32x32 RGB obrazových bodů s 1/16 multiplexova-

[1] Obě strany mohou přijímat i vysílat, avšak nikoli současně.



Obr. 1.13: Časové průběhy PPI rozhraní [10].

nými řádky, kterou adresuje čtyřbitová sběrnice. V jednom okamžiku jsou rozsvíceny pouze dva řádky, kde každý z řádků je ovládán řadiči typu MBO5024, tím je snížen počet nutných vodičů na třináct. Z toho šest vodičů nesou informaci o barevných kanálech, čtyři pro výběr adresy řádku a třech signálů pro ovládání řadiče.

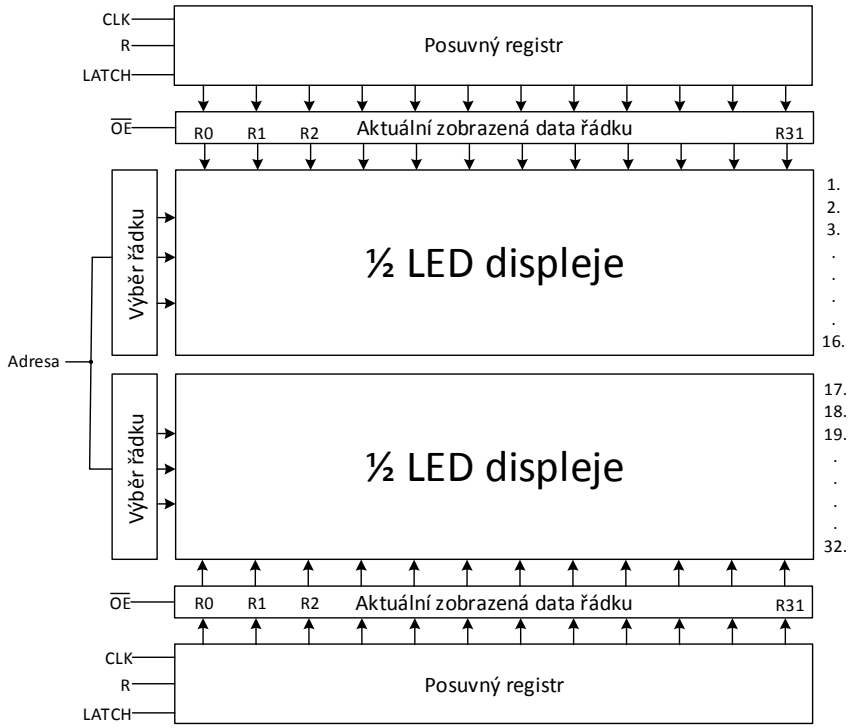
Panel má samostatné napájení. Je řízen 5 nebo 3,3 voltovými logickými úrovněmi, díky tomu není nutné řešit konverzi napěťových úrovní mezi FPGA a displejem. citeadafruit.

1.3.1 Princip ovládání displeje

Začneme rozborem jednoho světelného bodu (LED pixelu). Každý světelný bod může v základu zobrazit pouze tři barvy (červená, zelená a modrá) a jejich kombinace. Tyto barvy buď svítí, nebo nesvítí. Ve výsledku to znamená, že bez použití nějakého algoritmu, lze docílit maximálně sedmi barev na jeden světelný bod. Pro větší množství výběru barev je vhodné využít časového spínání barevných kanálů (pomocí pulsní šířkové modulace nebo binární kódové modulace).

Princip ovládání ovládání displeje vychází z obr. 1.14. Displej je rozdělen horizontálně na dvě poloviny. Aktuálně zobrazovaný řádek na obou polovinách je vybírán čtyřbitovém signálem (značený *ADRESA*), jehož hodnota přímo odpovídá zobrazovanému řádku. Například bude-li nabývat hodnotu „1000B“, bude rozsvícen osmý řádek první poloviny a zároveň osmý řádek druhé poloviny displeje.

Data k zobrazení jsou sériově posílána do posuvných registrů. Barvy jsou vedeny po 2x3 signálových vodičích, kde každá barva má vyhrazený jeden signálový vodič. První tři barvonosné vodiče jsou pro horní polovinu displeje a zbylé tři pro spodní část. Posuvný registr je součástí řadiče MBI5024, který je řízen hodinovými impulsy (značeno CLK), reagující na náběžnou hranu. Zobrazovaná data se tímto způsobem posunou o jeden sloupec dál, po celém řádku a přímo na svojí pozici. Počet náběžných hran signálu CLK se odvíjí od počtu sloupců v jednom řádku. V našem případě je to 32. Pokud bych jich tedy bylo více, nebo méně. Byly by data



Obr. 1.14: Principiální schéma LED displeje [1].

pro daný pixel posunutá o počet impulsů doleva, nebo doprava. To by ovšem mělo za následek nesprávnou funkci displeje.

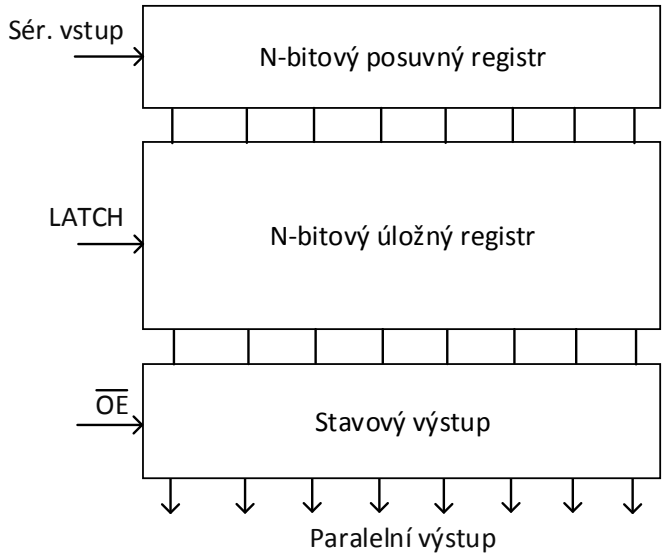
Další nezbytný signál pro správnou funkcionalitu displeje je signál *LATCH*, po jehož aktivaci překopíruje tyto naposouvaná data do speciálního úložného registru, kde tyto data zůstanou až do příchodu další aktivace tohoto signálu. Jako poslední zbývá pouze signál $\overline{OE}$, který aktivuje paralelní výstup úložného registru, na kterém jsou přímo připojeny diody. Pro lepší pochopení na obr. A.1 jsou zobrazeny jejich časové průběhy.

1.3.2 Řadič MBI5024

Řadič MBI5024 má jeden sériový vstup a šestnáctibitový paralelní výstup. Z toho plyne, že řídí maximálně šestnáct obrazových bodů jednoho barvonosného kanálu. Na obvodové desce se jich proto nachází dvanáct.

Vnitřní blokové schéma je naznačeno na obr. 1.15. Je složen z posuvného registru ovládaného náběžnou hranou hodinového impulsu, s maximální frekvencí 25Mhz. Dále se nachází úložný registr a stavový výstup, jehož princip byl vysvětlen výše.

Další důležité parametry, které jsou limitní pro tento řadič jsou v příloze (tab.



Obr. 1.15: Principiální schéma řadiče [3].

A.1), jehož časové průběhy naleznete také v příloze (obr. A.2).

1.3.3 Řízení jasu každého subpixelu

Barevná hloubka

Barevná hloubka vyjadřuje počet možných RGB odstínů, které obrazový bod může zobrazovat. Počet těchto odstínů vyjadřuje parametr BH (rovnice 1.1).

$$BH = 2^{R+G+B}[5]. \tag{1.1}$$

Parametry R, G a B popisují počet bitů na jednu barvu. Například když světelný bod RGB má barevnou hloubku 8 bitů na jednu barvu, tak potom může mít až šestnáct miliónů odstínů barev [5].

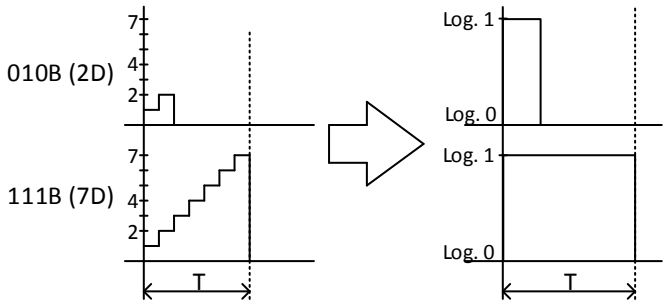
Implementace barevné hloubky: Jeden světelný bod je složen ze tří barevných diod (barevná, zelená a modrá), které svítí pouze na horní logickou úroveň. Lze tedy realizovat maximálně osm barevných odstínů (včetně černé). Pro větší barevnou hloubku je nutné použít pulsně šířkovou nebo binární kódovou modulaci.

Pulsně šířková modulace

Pulsně šířková modulace (obr. 1.16 vpravo) je technika, která se používá pro kódování velikosti signálu do šířky impulsu, jehož hlavním cílem je řídit sílu nebo

intenzitu elektrických zařízení. V našem případě to bude možnost měnit intenzitu svítivosti barevných kanálů, tím měnit barevnou hloubku.

Pulsní šířkovou modulaci si můžeme představit jako komparaci (obrázek vlevo) vstupního signálu (velikosti signálu barevného kanálu) s lineárně se zvyšujícím čítačem v čase. Pokud je tento čítač nižší, než je hodnota signálu barevného kanálu, výstupní signál je roven horní logické úrovni. Při zvýšení tohoto čítače nad stanovenou podmínku je výstupní hodnota rovna nule, po zbytek periody.



Obr. 1.16: Principiální průběhy PWM.

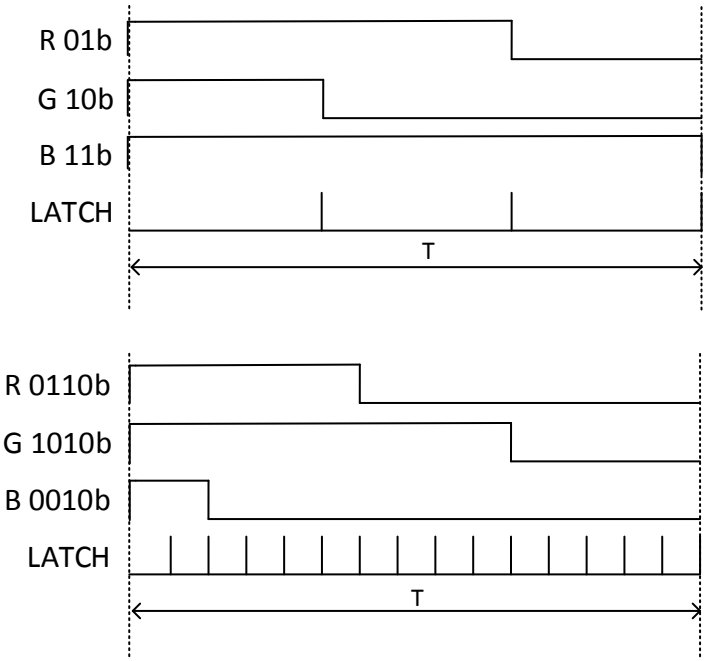
Použití PWM: Na obr. 1.17 jsou znázorněny časové průběhy. Horní obrázek je pro dvoubitovou barevnou hloubku a spodní pro čtyřbitovou. Z předchozího popisu víme, že odstín je udáván šířkou impulsu jednotlivých barevných kanálů. Zvolíme-li dvoubitovou barevnou hloubku, tak potom počet možných odstínů jednoho kanálu je roven čtyřem. (Černá barva se vyobrazí jako nulová úroveň celé periody).

Implementací barevné hloubky se při každé možné změně odstínu změní obrazová data do posuvného registru tak, aby při překopírování signálem *LATCH* měli požadovanou šířku.

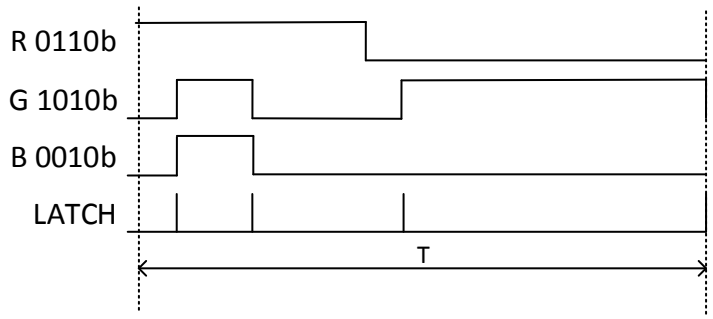
Binární kódová modulace

Binární kódová modulace, neboli BCM je v podstatě exponenciální vyjádření pulsně šířkové modulace. Význam má tedy stejný. S rozdílem, že perioda signálu není rozdělena na díly o stejné velikosti, ale exponenciálně (viz obr. 1.18), kde každý díl je mocnina binární hodnoty, kterou nabývá. Velikost následujícího dílu je vždy dvojnásobkem předchozího. Touto metodou dojde ke snížení počtu přenesených řídících dat [14].

Na tomto obrázku můžete vidět, že v daném čase může být jiná logická úroveň, ale celková šířka zůstává stejná.



Obr. 1.17: Časové průběhy barevné hloubky pomocí PWM.

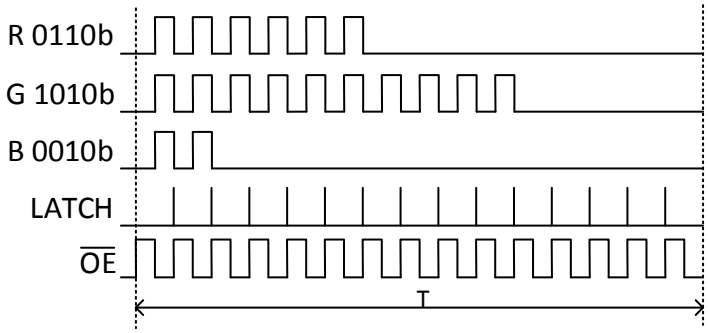


Obr. 1.18: Časové průběhy barevné hloubky pomocí BCM.

1.3.4 Řízení globálního jasu displeje

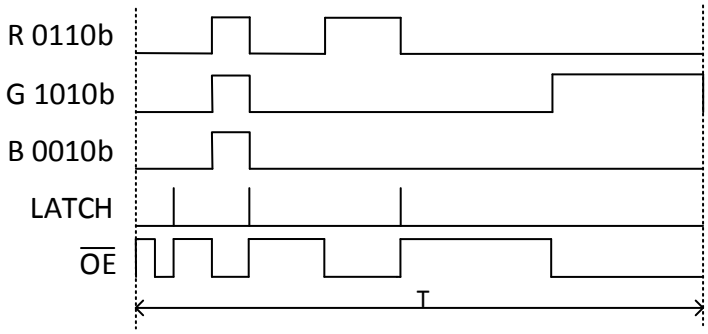
Implementace globálního jasu bude podobná jako u barevné hloubky. Buď s pomocí PWM nebo BCM. S rozdílem, že bude ovládán signálem $\overline{OE}$, který v podstatě na krátkou dobu vypne displej. Tím doje ke snížení jasu.

Chceme-li, aby například jas s použitím PWM byl 50%. Pro čtyřbitový signál bude řídící signál $\overline{OE}$ v každé z šestnácti částí polovinu času v log. 1 a polovinu času v log. 0 (viz obr. 1.19).

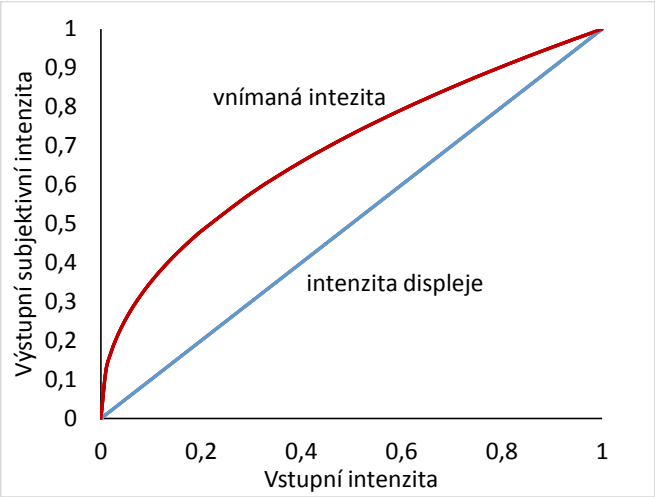


Obr. 1.19: Časové průběhy ovládání globálního jasu pomocí PWM.

Chceme-li, aby naopak globální jas byl řízen pomocí BCM, musí se přepočítat čas, kdy bude displej zapnutý, protože doba zapnutí je řízena exponenciálně. (viz obr. 1.20).



Obr. 1.20: Časové průběhy ovládání globálního jasu pomocí BCM.



Obr. 1.21: Charakteristika závislosti vstupní na výstupní intenzitě bez použití gamma korekce.

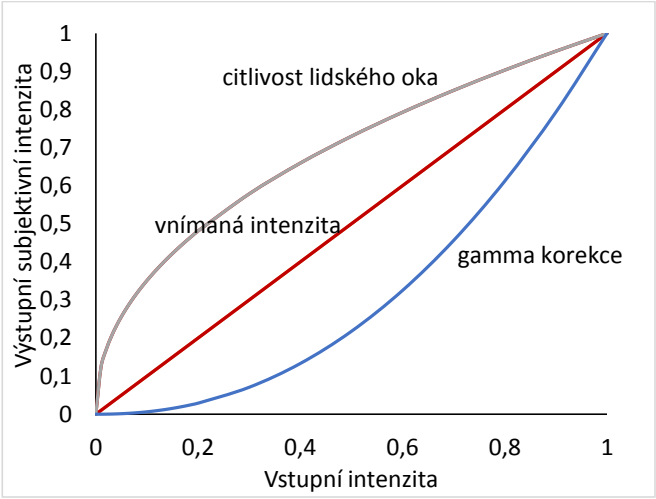
1.3.5 Gamma korekce

Lidské oko je nelineárně závislé na intenzitě světelného záření. To má za následek, že s lineární změnou intenzity světla lidské oko nevnímá změnu jako lineárně se měnící, ale nelineárně. Z tohoto důvodu se používá tzv. gamma korekce, která tuto nelinearitu kompenzuje. K výpočtu gamma korekce se používá vzorec 1.2, kde vstupní jas pixelu ($I_{vstupni}$) je intenzita světla bez použití gamma korekce. Výstupní jas pixelu ($I_{vystupni}$) udává skutečnou hodnotu po použití gamma korekce a „Gamma“ udává charakteristiku intenzity světla. Obvykle nabývá hodnot 1.0 - 2.2 [4].

$$I_{výstupní} = I_{vstupní}^{Gamma}[4]. \tag{1.2}$$

Na grafu 1.21 jsou zobrazeny charakteristiky bez použití gamma korekce, z kterého je patrné, že vnímaná subjektivní intenzita je nelineární. I přes lineární intenzitu displeje lidské oko zpočátku vnímá zvyšující se intenzitu velmi rychle a poté pouze pozvolna roste.

Zatímco s použitím gamma korekce (gamma = 2,2) výsledná vnímaná intenzita působí lineárně (viz obr. 1.22).



Obr. 1.22: Charakteristika závislosti vstupní na výstupní intenzitě s použitím gamma korekce.

1.4 LED panel firmy Ing. Ivo Herman, CSc.

Jedná se o panel o rozměrech 32x20 obrazových bodů nebo 32x30, který může být rozšířen o další panely. Maximálním rozlišení všech panelů je však 320x240 bodů.

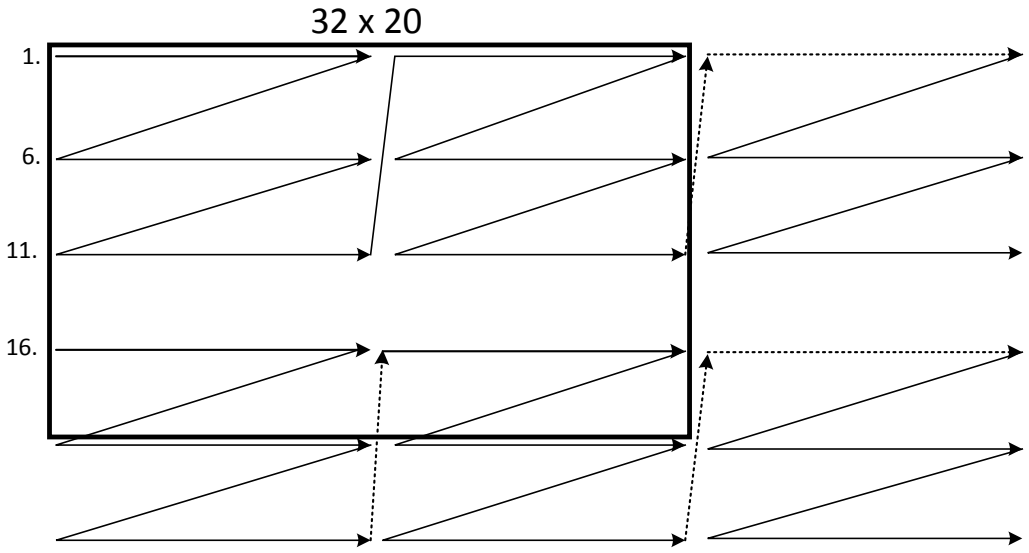
Pro panel s rozlišením 32x20 je ovládání panelu úplně shodné s panelem o rozměrech 32x30. Body, které v panelu s nižším rozlišení chybí, jsou nahrazeny pouze posuvnými registry.

Každý tento panel časově multiplexuje pět řádků. Panel je řízen společným $\overline{OE}$ signálem a $LATCH$ signálem. Obdobně, jak to bylo u displeje Adafruit. Dále je zde připojeno pouze šest barvonosných vodičů, jehož princip bude vysvětlen níže.

1.4.1 Princip ovládání panelu

Princip ovládání panelu je podobný s ovládáním displeje Adafruit s rozdílem, že multiplexuje pět řádků (obr. 1.23), který je vybírán tříbitovou adresou. Z toho vyplývá, že by bylo potřeba dvanáct barvonosných vodičů na každý pane. To je vyřešeno tím, že posuvný registr řadiče, který by měl zobrazovat první řádek, je naplněn třemi řádky (místo 32 obrazovými body, je 96). Z obrázku je tedy patrné, že za předpokladu výběru první adresy, je posuvný registr zpočátku plněn šestnácti hodnotami prvního řádku, poté je plněn šestnácti hodnotami šestého řádku, pilovitě až do jedenáctého řádku. Následně dojde k přesunu na první řádek a postup je opakován. Šestnáctý řádek je řešen dalším stejně dlouhým posuvným registrem. Tím je docíleno, že je potřeba pouze šest barvonosných vodičů. Rozšířením šířky (přidáním

dalšího panelu) dojde pouze ke zvětšení posuvného registru na dvojnásobek, nikoli zvýšení počtu vodičů.



Obr. 1.23: Princip ukládání dat do posuvného registru LED panelu Ing. Ivo Herman, CSc..

Panel může zobrazit pouze 8 barev, proto je nutné pro zvýšení barevné hloubky opět použít PWM nebo BCM.

1.4.2 Zapojení panelů

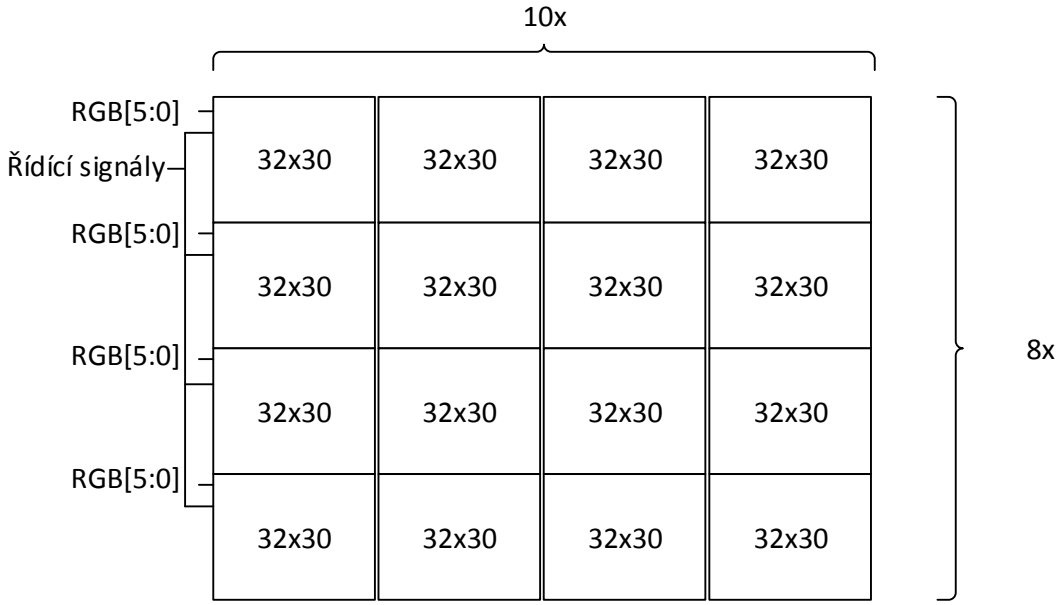
Zapojení panelů vychází z obr. 1.24. Zvětšením šířky rozlišení (přidáním dalšího panelu) nedochází k nárůstu počtu vodičů přivedených od FPGA, ale naroste počet hodinových impulsů, který je přiváděn do posuvných registrů. Obrazová data jsou v podstatě sériově přeposouvána na správnou pozici dále do dalších panelů.

Zvýšení šířky rozlišení se počet barvonosných vodičů poměrně zvyšuje. Řídící vodiče jsou pro všechny panely společná, proto není potřeba více vodičů.

Maximální počet potřebných vodičů pro ovládání panelu s maximálním rozlišení je roven součtu všech barvonosných a řídících kanálů, tedy 54 (8 panelů 32x30 na výšku -> 8 x 6 barvonosných vodičů + 6 řídících)

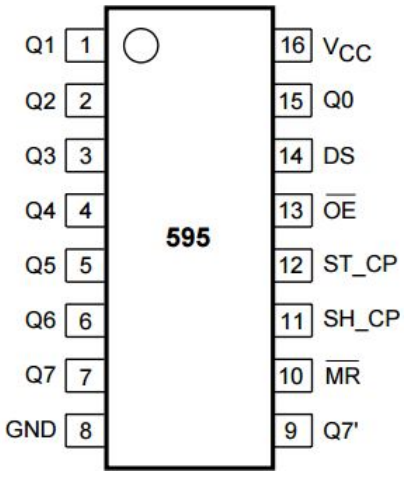
1.4.3 Řadič 74HC595

Základem panelu jsou 8-bitové LED řadiče 74HC595 1.25, který má osm paralelních výstupů, jeden sériový vstup a výstup, a další tři řídící vodiče. Řadič plní obdobnou



Obr. 1.24: Zapojení LED panelů firmy Ing. Ivo Herman, CSc..

funkci s řadičem Adafruit [8].



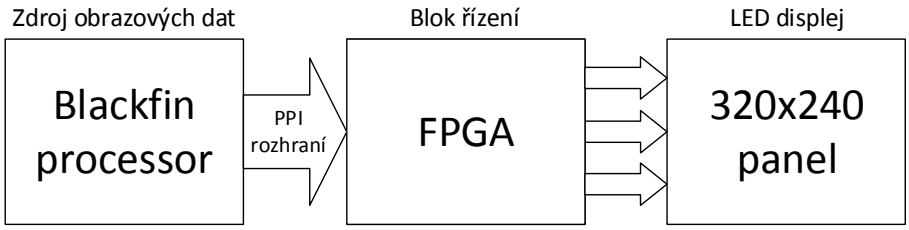
Obr. 1.25: Blokové schéma obvodu 74HC595.

Tab. 1.2: Význam pinů obvodu 74HC595 [8].

Číslo pinu	Symbol	Popis
1-7	Q1-Q7	Paralelní výstup
8	GND	Uzemnění
9	Q7'	Sériový výstup
10	MR	Reset
11	SH_CP	Hodiny posuvného registru
12	ST_CP	Překopírování do úložného registru
13	$\overline{OE}$	aktivace výstupu
14	DS	Sériový vstup
15	Q0	Paralelní výstup
16	Vcc	Napájení obvodu

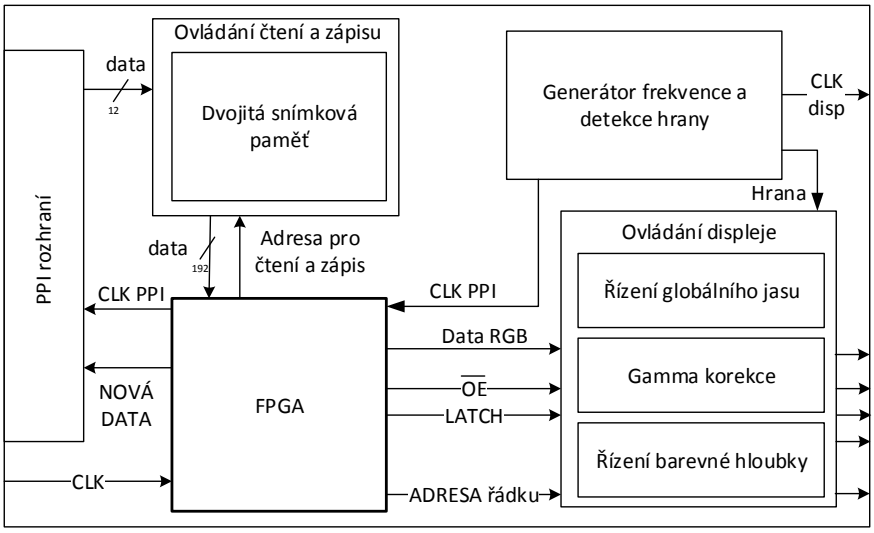
2 NÁVRH VHDL MODULU

Tato část se věnuje návrhu VHDL modulu pro FPGA, který řídí ukládání toku dat procesoru Blackfin do snímkové paměti, a tyto data budou zobrazeny na LED panelu firmy Ing. Ivo Herman, CSc.. Tento návrh představuje blokové schéma na obr. 2.1.



Obr. 2.1: Blokové schéma řešení.

Na obrázku 2.2 se nachází vnitřní schéma bloku řízení. Nejdůležitější blok představuje FPGA. Ovládá komunikaci periferního rozhraní PPI mezi zdrojem dat a dvojitou snímkovou pamětí do míst, kam jsou data ukládána. FPGA přímo tedy nečte data z procesoru, ale ze snímkové paměti. Dále je zde blok pro ovládání displeje. Typicky obsahuje řídící kanály a také signál, který nese informaci o aktuálním RGB hodnotám právě jednoho cyklu posuvného registru.



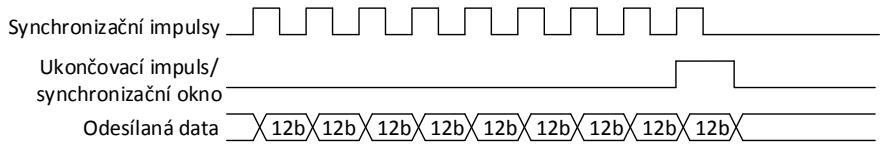
Obr. 2.2: Blokové schéma bloku řízení.

Tab. 2.1: Význam signálů v bloku řízení.

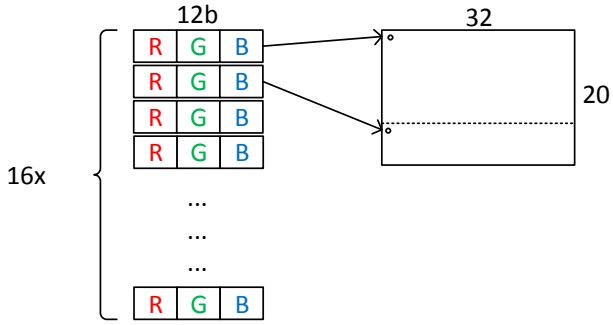
Signál	Význam
Data [11:0]	Zapisovaná data do blokové paměti
Data [191:0]	Čtená data z blokové paměti
Clk_PPI	Synchronizační impulsy pro zdroj dat
Clk	Dedikované hodinové impulsy
Adresa řádku	Adresa multiplexovaného řádku
LATCH	Překopírování dat z posuvného registru do úložného registru
$\overline{OE}$	Aktivace displeje
CLK disp	Hodiny posuvného registru
Adresa pro čtení a zápis	Vyplývá z názvu
Hrana	Signál nesoucí informaci o nástupné hraně řízených hodin
Data RGB	Buffer barvonosných kanálů

2.1 Rozhraní PPI

Procesor Blackfin generuje obrazová data, která jsou odesílána po PPI rozhraní (viz obr. 2.3). Blok řízení (FPGA) vygeneruje každých 40ms (odpovídá 25 snímků/s) hodinové impulsy o celkovém počtu bodů jednoho obrazového snímku. Na tyto impulsy reaguje zdroj dat a při každé náběžné hraně tohoto impulsu odešle zdroj dat data pro jeden obrazový bod (12 bitů), nezávisle na velikosti barevné hloubky. Na posledním impulsu souběžné hrany následuje ukončovací impuls, který plní funkci synchronizačního okna (popsané v teoretickém úvodu). Tento ukončovací impuls tedy zdroj dat informuje, že má nachystat nová data.



Obr. 2.3: Navrhovaná komunikace mezi zdrojem dat a FPGA.



Obr. 2.4: Principiální schéma čtení dat z BRAM.

2.2 Dvojitá snímková paměť

Bloková RAM paměť je bude pomocí IP Core generátoru jako Simple Dual port RAM s dvanáctibitovou šířkou zápisu, která je rovna jednomu přenesenému obrazovému bodu. Šířka čtení je však rozdílná. Je 192 bitů dlouhá (obr. 2.4) a nese datovou informaci o aktuálně přenášených bodech odesílaného do posuvných registrů pro jeden sloupec.

2.2.1 Čtení a zápis

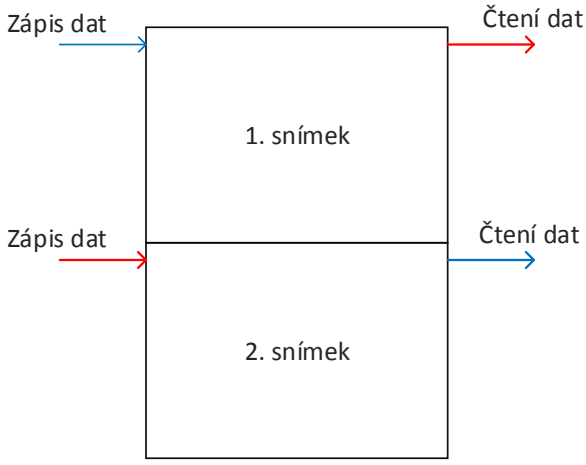
Generovaná data, která vstupují do paměti jsou obvykle odesílána po celých řádcích. Problém nastává tehdy, kdy jsou tyto data čtena. Protože jsou data čtena vcelku (po 192b), která reprezentují sloupec bodů (zároveň jednoho cyklu posuvného registru), musí být nejprve data transponována. To je docíleno přeskládáním, buď přímo procesorem Blackfin nebo změnou adresy při ukládání každého obrazového bodu.

Velikost celé paměti odpovídá dvojnásobku maximálnímu rozlišení, tedy 320x240 bodů. Výška paměti je odlišná a záleží, zda-li je brána z pohledu čtení, nebo zápisu.

Snímek je zobrazován přibližně 200Hz, zatímco nový snímek je vyžádán každých 25Hz. Čtení dat je 8x častější, než je zápis. Mohlo by docházet ke kolizím dat, proto je zde dvojitá snímková paměť, aby nedocházelo k přepisování aktuálně zobrazovaného snímku daty nového snímku.

Na obr. 2.5 je zobrazeno rozdělení této paměti. V modulu je implementován algoritmus, který přepíná mezi těmito snímky. V první fázi (značeno červenou barvou) panel zobrazuje data horní poloviny paměti (1. snímek), zatímco probíhá zápis dat do spodní poloviny paměti (2. snímek). Nová data do paměti jsou odesílána každých 40ms a poté, kdy jsou všechna data nového snímku zapsána do paměti, modul začne

číst adresu paměti spodní poloviny (značeno modrou barvou), které se zobrazuje na panelu do doby, než jsou nahrána nová data do vrchní části paměti.



Obr. 2.5: Schéma uspořádání dat dvojité snímkové paměti.

2.3 Generátor frekvencí

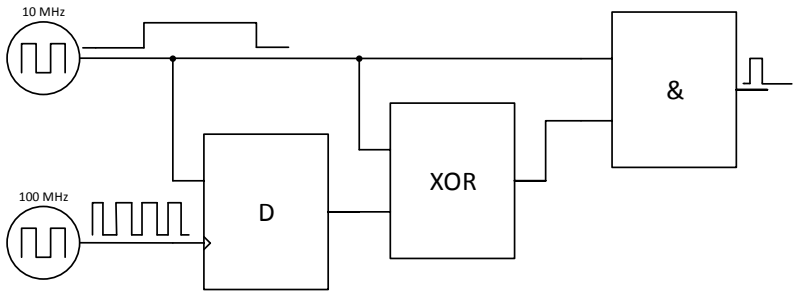
Blok generátor frekvence generuje hodinové impulsy pro zdroj obrazových dat pro PPI rozhraní a zároveň hodinové impulsy pro ovládání displeje, resp. řízení posuvných registrů. Tyto hodinové impulsy nejsou generovány celou dobu, ale právě v okamžik, když je to vhodné. Jsou proto spínány.

S tímto generátorem frekvencí je úzce spjata jejich detekce náběžné hrany, která dále nese informaci o aktuálním počtu hodinových impulsů synchronizace zdroje dat, ale také i informaci počtu bodů odeslaných do řadiče.

Detekce hrany je vytvořena za pomocí funkce XOR, logického součinu AND, společně s aktuální a časově zpožděném signálem frekvence a s výchozí frekvencí FPGA podle schématu obr. 2.6 (řízená frekvence 10MHz, vstupující do klopného obvodu D, logického obvodu XOR a AND je pouze názorná), jehož pravdivostní tabulku naleznete v tab. 2.2.

2.4 Ovládání displeje

Ovládání displeje je navrženo podle Moorova stavového automatu. Výhoda, spočívá v jednoduchosti implementace, avšak jeho reakce na vstupy je zpožděná o jeden hodinový cyklus, než je s použitím Mealyho stavového automatu.



Obr. 2.6: Blokové schéma detekce hrany.

Tab. 2.2: Pravdivostní tabulka pro detekci hrany.

Aktuální	Zpožděná	Akt. XOR Zp.	(Akt. XOR Zp.) AND Akt.
0	0	0	0
1	0	1	1
0	1	1	0
1	1	0	0

2.4.1 Přetransformování obrazových dat

Nejprve bude popsán formát obrazových dat, který do tohoto stavového automatu vstupuje.

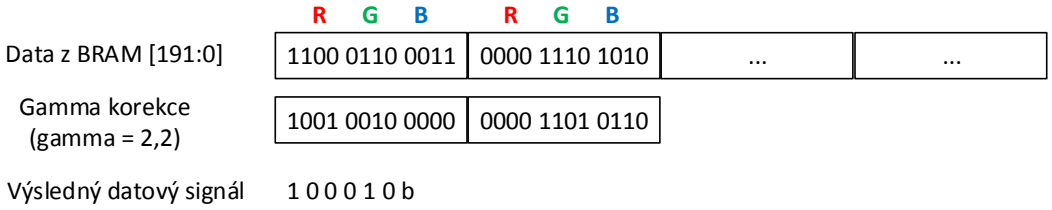
Vstupní obrazová data jsou v bloku řízení reprezentována signálem „Data“, která přímo neodpovídají obrazovým bodům ze snímkové paměti, ale jsou už přepočítána tak, aby mohla být přímo odeslána do řadiče (obr. 2.7). Na každou barvu je aplikována gamma korekce a poté je z každého barevného kanálu vybrán významný bit, který odpovídá barevné hloubce v daném časovém okamžiku, které bude popsáno dále. Tímto způsobem je redukovaná délka dálka dat na aktuálně potřebné informace. Ze 192b dlouhého signálu lze snížit délku dat (s nejnižším rozlišením) až na šest bitů.

Gamma korekce a vytvoření tabulek

Přepočet gamma korekce probíhá podle rovnice 1.2, kterou si ukážeme na příkladu.

Příklad: R 1100b = $(12)_{10}$, gamma = 2,2

$$I_{výstupní} = I_{vstupní}^{Gamma} = (12/15)^{2.2} = 9,189 = 1001b. \tag{2.1}$$

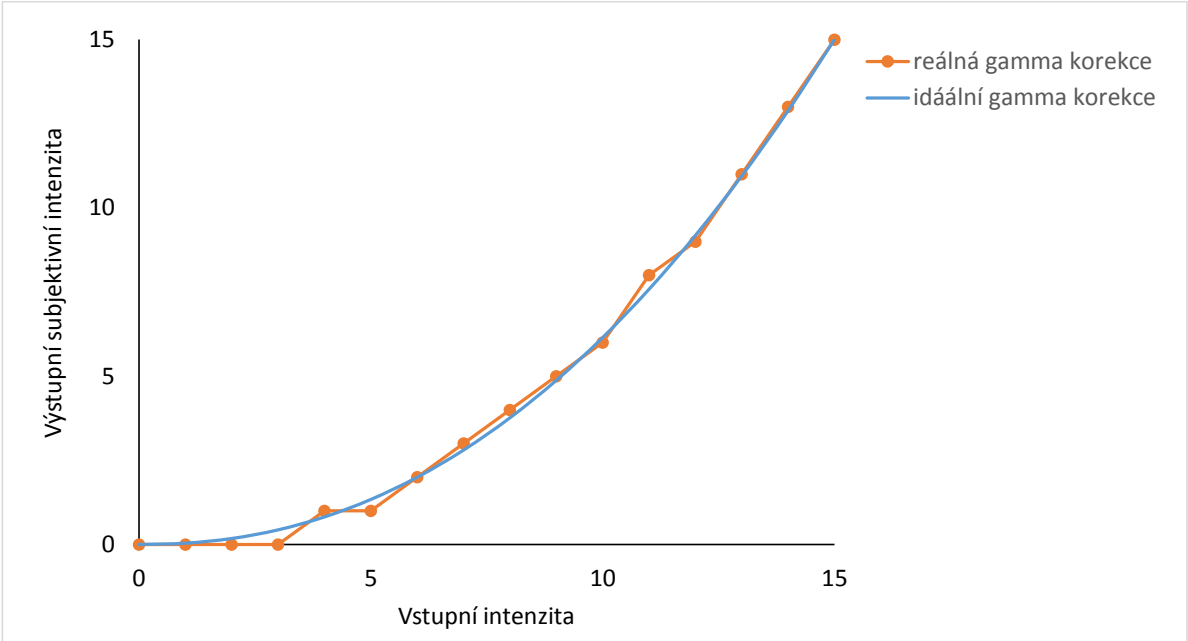


Obr. 2.7: Princip přetransformování dat z BRAM do posuvného registru.

Z předchozího příkladu je patrné, že dochází k zaokrouhlování hodnot na celá čísla, která jsou přepočítána zpět do binární hodnoty o velikosti barevné hloubky.

Při syntéze programu jsou poté přepočítány korekční gamma tabulky, jejich přesnost je závislá na použité barevné hloubce (viz tab. 2.3)

Na obr. 2.8 jsou znázorněné charakteristiky gamma korekce. Ideální (modrá charakteristika) a reálná (oranžová charakteristika) gamma korekce s čtyřbitovou barevnou hloubkou jednoho sub-pixelu.



Obr. 2.8: Charakteristiky závislosti vstupní na výstupní intenzitě ideální gamma korekce a reálné gamma korekce s čtyřbitovou hloubkou na jednu barvu.

Z charakteristik je patrné, že vstupní intenzita o hodnotě 3, je chyba gamma korekce nejvyšší. Výstupní intenzita je oproti ideální subjektivní intenzitě stále rovna nule, která je rovna přibližně 0,4.

Tab. 2.3: Tabulka pro výpočet reálné výstupní intenzity, gamma = 2,2.

Vstupní intenzita	Ideální výstupní intenzita	Reálná výstupní intenzita
0	0	0
1	0,038787	0
2	0,17822	0
3	0,434868	0
4	0,818884	1
5	1,337903	1
6	1,998128	2
7	2,804828	3
8	3,762604	4
9	4,875554	5
10	6,147386	6
11	7,581487	8
12	9,180984	9
13	10,94878	11
14	12,8876	13
15	15	15

Barevná hloubka

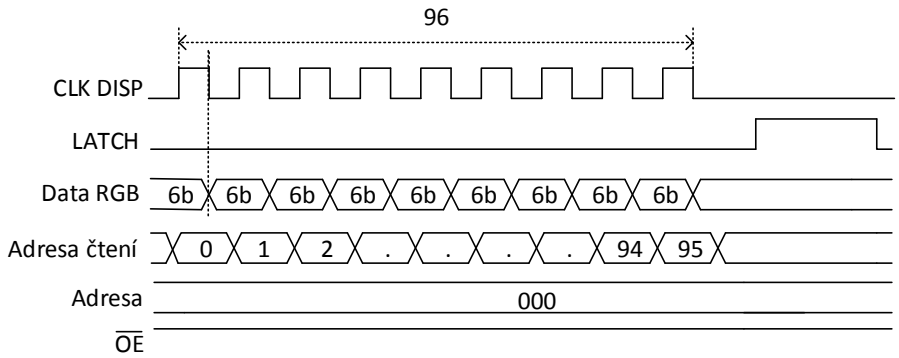
Barevná hloubka byla implementována za pomocí BCM. Je tedy zřejmé, že je v daný časový okamžik použit právě jeden významný bit. Výsledný datový signál se redukuje na bity, které jsou ve vhodný okamžik právě odesílané přímo do řadičů.

2.4.2 Základní postup ovládání displeje s implementací BCM

Zde bude popsán postup ovládání displeje s rozlišením 32x20 bodů a čtyřbitovou barevnou hloubkou na jeden sub-pixel.

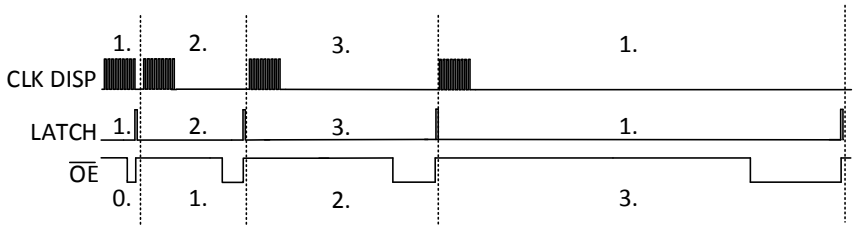
1. **Předpřipravení obrazových dat:**
   Tím je myšleno odeslání obrazových dat jednoho multiplexovaného řádku do posuvných registrů a překopírování těchto dat do úložného registru do doby, než dojde k první aktivaci displeje. Obrazová data jsou vždy předpřipravena v úložném registru a jsou aktivována ve vhodný okamžik.
   Pro lepší pochopení jsou na obr. 2.9 znázorněny časové průběhy. Signál *Data RGB* nese informaci o aktuální hodnotě barvonosných kanálů nultého řádu barevné hloubky, která je přetransformována podle obr. 2.7. Data jsou vždy měněna na sestupnou hranu synchronizačního signálu posuvného registru ře-

diče (značeného CLK DISP) tak, aby na náběžné hraně již byly platné. Adresa čtení z blokové paměti je naopak měněna při nástupné hraně toto synchronizačního signálu, aby před sestupnou hranou byla data přetransformována. Tyto data představují právě signál *Data RGB*. Počet hodinových impulsů signálu CLK DISP je roven součtu bodů třech řádků, tedy 96. Po tomto kroku jsou data v posuvném registru na správných pozicích. Nezbývá nic jiného, než je signálem *LATCH* překopírovat do úložného registru.



Obr. 2.9: Teoretické časové průběhy na vstupu displeje při předpřipravení dat.

2. **Zobrazení jednoho multiplexovaného řádku:** Zobrazení jednoho řádku vychází z průběhů, jež jsou zobrazené na obr. 2.10. V tuto chvíli jsou v úložném registru řadiče uloženy hodnoty bodů pro nultý řád barevné hloubky.
   Na začátku průběhu je v krátké době aktivován logickou nulou výstup těchto úložných registrů, který na displeji zobrazí aktuálně nultý řád barevné hloubky, zatímco do posuvných registrů je posílán první řád barevné hloubky.
   Před koncem tohoto intervalu je překopírován do úložného registru, protože pokud by překopírování došlo na začátku intervalu, výstup displeje by zobrazoval další řád, který momentálně nechceme.
   V dalším kroku je časový interval aktivovaného displeje dvojnásobný, který koresponduje s právě aktivním řádem barevné hloubky.
   V poslední části časových průběhů jsou do posuvných registrů odesílána obrazová následujícího řádku, aby se v dalším kroku zobrazovala data následujícího řádku, nultého řádu.
3. **Opakování bodu 2**



Obr. 2.10: Teoretické časové průběhy na vstupu displeje při zobrazování jednoho multiplexovaného řádku.

2.4.3 Popis Stavového automatu

Navržený stavový automat vychází ze přechodového diagramu (viz obr. 2.11). Následně zde budou vysvětleny všechny stavy, ve kterých se může stavový automat nacházet.

Stav st_idle

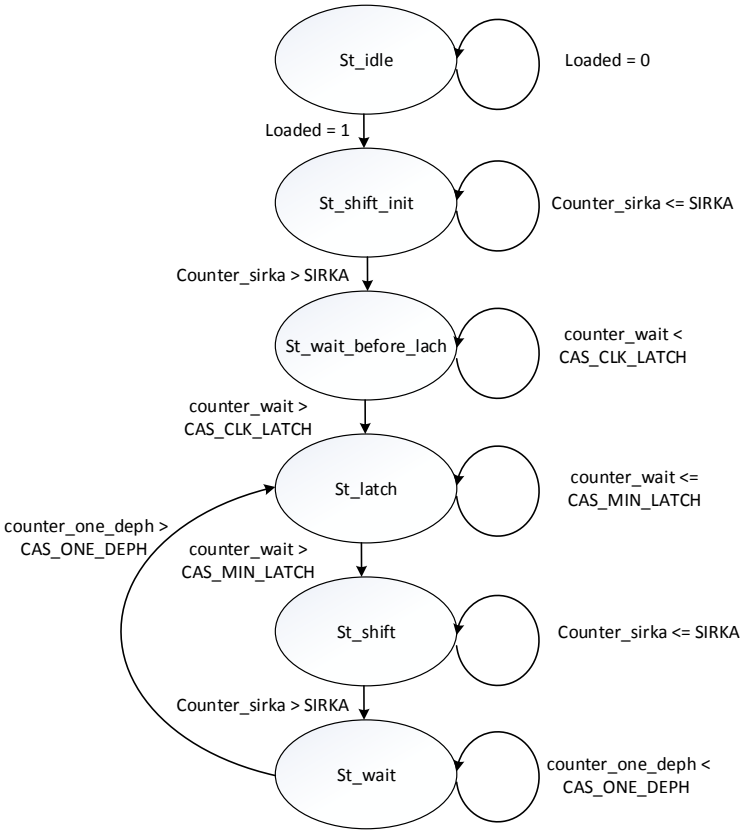
Inicializace všech proměnných na výchozí hodnoty. Stavový automat čeká do načtení prvního snímku (značeno loaded). Poté přechází do dalšího stavu.

Tento stav má za úkol tyto funkce:

- Vynulování adres ukazující do blokové paměti. Přechod na první snímek v paměti.
- Vypnutí displeje.
- Resetování proměnné pro implementaci barevné hloubky.
- Přesun na první multiplexovaný řádek.
- Resetování čítačů, které jsou nezbytné pro vytváření časových úseků.
- Resetování ostatních proměnných.

Stav st_shift_init

Před zobrazením prvního multiplexovaného řádku je potřeba nahrát obrazová data do posuvného registru řadiče a to spuštěním generátoru frekvence. Obrazová data musí být s těmito impulsy synchronizována na náběžnou hranu. Před každou nástupní hranou je nejprve nastavena správná adresa snímku v BRAM. Načtená obrazová data jsou přetransformována do správného formátu (viz kapitola 2.2) a odeslána na výstup barvonosných signálů. Transformace dat probíhá nezávisle na stavovém automatu. Je závislá na čítači counter_sirka, který vyjadřuje počet obrazových bodů odeslaných do řadiče. Dále je závislá na aktuální platné pozici bitu, která implementuje barevnou hloubku, a také na aktuálně vybranými multiplexovanými řádky.



Obr. 2.11: Stavový diagram ovládání displeje.

Po naplnění všech posuvných registrů stavový automat přechází do stavu st_before_latch.

Stav st_before_latch

Stav st_before_latch plní pouze za funkci časové pomlky (značeno counter_wait) mezi naposouváním dat v posuvném registru a překopírováním do úložného registru v řadiči.

Stav st_latch

Ve stavu st_latch se data překopírují do úložného registru aktivací signálu *LATCH* a určitý čas se počká, aby řadič na tento signál mohl zareagovat.

Stav st_shift

Tento stav je podobný jako stav st_shift_init. Odešle do posuvného registru řadiče obrazová data. Zároveň signálem $\overline{OE}$ aktivuje výstup úložného registru na displeji,

podle úrovně nastaveného jasu. Avšak by musela být nastavena na velmi vysokou úroveň jasu. Nižší úrovně jasu implementuje další stav.

Stav st_wait

Poslední stav ovládání displeje je stav st_wait, který čeká požadovaný čas v závislosti na aktuálnímu řádu zobrazované barevné hloubky. Například pokud by doba trvání nultého řádu barevné hloubky byla „t“ a byl by právě zobrazovaný druhý řád barevné hloubky, byla by tato doba rovna „4t“. Další funkcí tento stav plní, jak již bylo zmíněno, implementace nižší úrovně jasu. Poslední funkcí toto stavu je zjišťovat, jaký multiplexovaný řádek má právě zobrazovat a ten odesílat je signálem *ADRESA* do displeje.

V tuto chvíli jsou již předpřipravená data v posuvném registru řadiče a po ukončení probíhajícího stavu stavový automat přechází do stavu st_latch a celý proces se opakuje.

3 VÝSLEDKY DIPLOMOVÉ PRÁCE

V této části budou popsány průběhy teoretické (simulace) a průběhy reálné celé diplomové práce. Konkrétně komunikaci na rozhraní PPI mezi zdrojem dat a FPGA, ale také i komunikaci mezi FPGA a panelem firmy Ing. Ivo Herman, CSc. o rozměrech 32x20.

3.1 Rozhraní PPI

Simulace (obr. 3.1) odpovídá teoretickým průběhům na rozhraní PPI. Na obrázku jsou znázorněny průběhy synchronizačních impulsů (značeno clk_ppi), ukončovacího impulsu a měnící se adresy, která ukazuje na část blokové paměti v oblasti druhého snímku. První snímek v paměti je v tuto chvíli čten.

Obr. 3.1: Simulace komunikace na rozhraní PPI.

Reálné průběhy, které byly viděny na osciloskopu (viz obr. 3.2) odpovídají simulaci. Vrchní průběh tvoří synchronizační impulsy. Na konci posledního impulsu je ukončovací impuls (prostřední průběh), aby bylo možné synchronizovat zdroj dat s FPGA. Spodní průběh vyjadřuje nikoliv adresu zápisu (ta je řízena pouze v FPGA), ale obrazová data na tomto rozhraní.

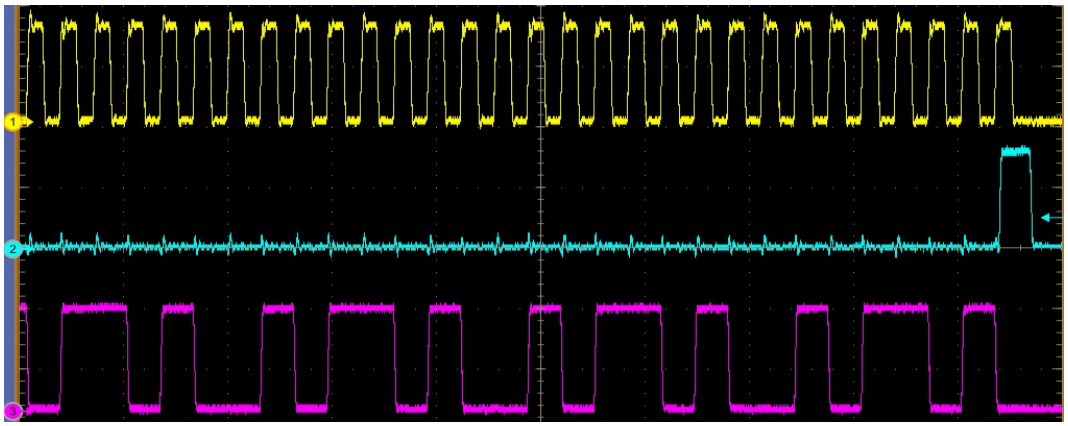
Z průběhů osciloskopu je patrné, že každý obrazový bod je odesílán na náběžnou hranu synchronizačního impulsu.

3.2 Ovládání displeje

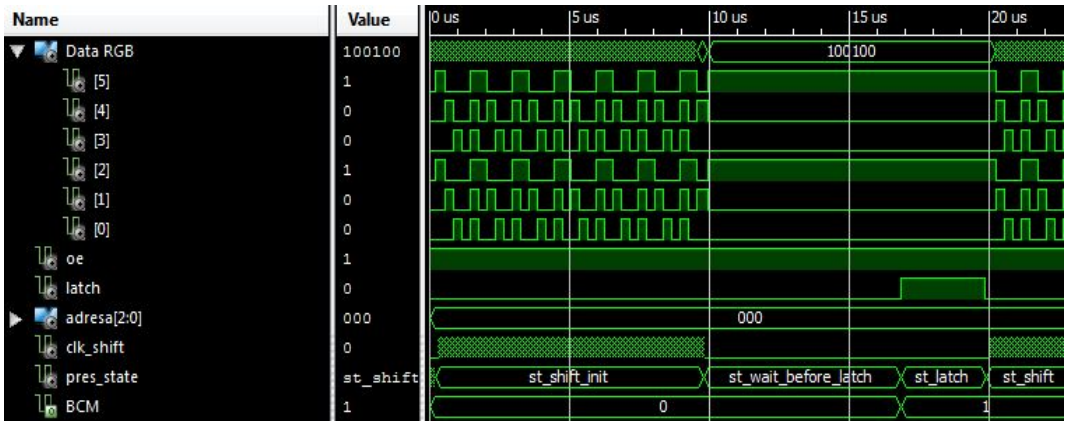
Na obr. 3.3 je vyobrazena simulace průběhů z kap. 2.4.2. prvního bodu, tedy předpřipravení obrazových dat. V simulaci je také vidět, jak stavový automat přechází do jednotlivých stavů, ve které vykonává dílčí funkce.

Simulaci druhého bodu popisuje obr. 3.4. Na této simulaci je patrná implementace barevné hloubky a globálního jasu za pomocí BCM. Signál BCM popisuje číslo řádu barevné hloubky, který je do řadiče odesílán.

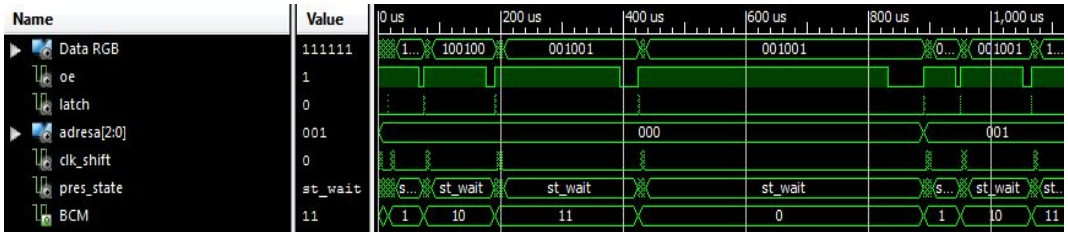
Reálné průběhy, které byly viděny na osciloskopu (viz obr. 3.5) se podobají teoretickým průběhům simulace. Na tomto snímku odpovídá simulaci druhého bodu. Tyto průběhy korespondují třem signálům, které vstupují do řadiče panelu. *LATCH*



Obr. 3.2: Reálné průběhy komunikace na rozhraní PPI (vrchní - synchronizační impulsy, prostřední - ukončovací impuls, spodní - posílaná data).



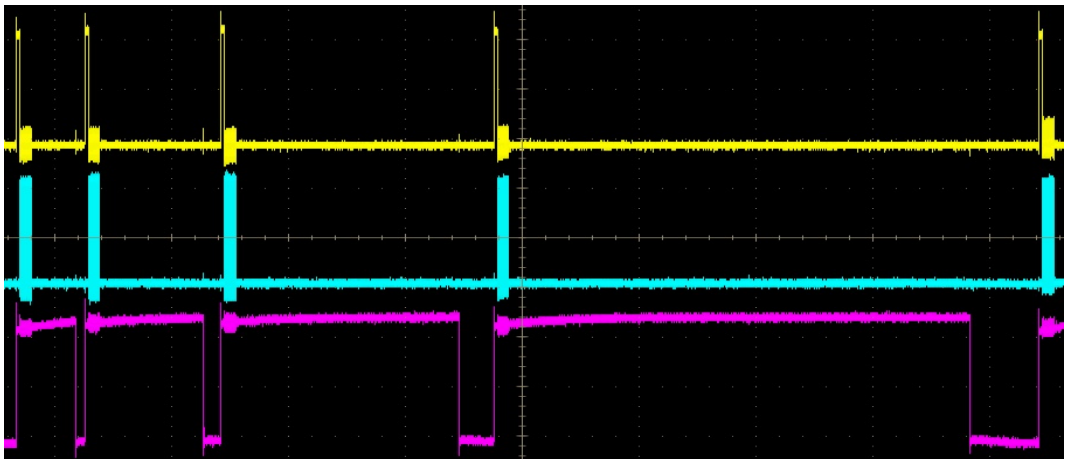
Obr. 3.3: Simulace průběhů stavového automatu pro stav st_shift_init, st_wait_before_latch a st_latch.



Obr. 3.4: Simulace po průběhů po implementaci barevné hloubky.

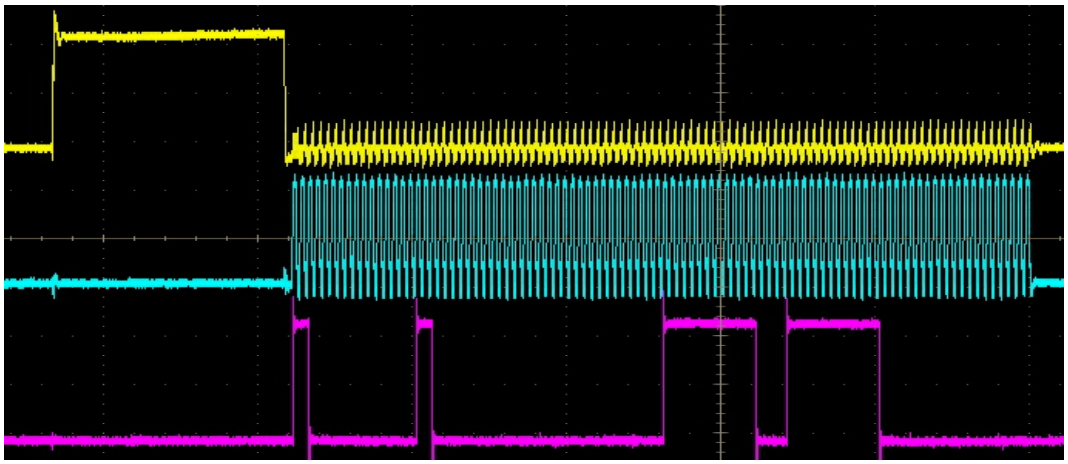
(vrchní), hodinové impulsy posuvného registru (prostřední) a signál $\overline{OE}$ (spodní). hodinové impulsy jsou přiblíženy na obr. 3.6, jehož spodní průběh není signál $\overline{OE}$, ale data barvonosného kanálu.

Při dostatečném přiblížení bylo patrné, že signál vstupující do posuvného registru nemá tvar obdélníkového, ale spíše se podobal sinusovému. Tento fakt může



Obr. 3.5: Reálné průběhy vstupující do displeje (vrchní - signál *LATCH*, prostřední - hodinový signál, spodní - signál $\overline{OE}$).

mít vliv na prodloužení náběžné hrany. Na osciloskopu se objevovaly různé přeslechy. Ty mohly způsobit například krátkodobý pokles napětí náběžné hrany hodin posuvného registru, tím nepříznivě ovlivnit činnost posuvného registru a vyvolat dvojnásobné aktivování posuvného registru, čím by se posunula data barvonosného kanálu na chybnou pozici. Problém by mohl vyřešit spínání toho signálu s vyšší frekvencí, než je výchozí frekvence FPGA a nebo generovat hodinové signály bez použití bloku „Generátoru frekvence“ jiným způsobem.



Obr. 3.6: Reálné průběhy hodin vstupujícího do posuvného registru (vrchní - signál *LATCH*, prostřední - hodinový signál, spodní - data).

3.2.1 Zhodnocení modulu

Zhodnocení modulu popisuje tvz. „summary report“, zde jsou popsány informace o využitých hardwarových prostředcích, při programování tohoto modulu, které bylo vytvořeno při mapování obvodu.

Využití hardwaru

Jedná se o množství využitého hardwaru pro naprogamování modulu. V tabulce 3.1 jsou informace bez použití register balancingu a v tab. s register balancingem. (viz tab. 3.2)

Tab. 3.1: Tabulka využití hardwarových prostředků bez použití register balancingu.

Název	Využito	Celkem	Poměr
Počet Slice Registrů	142	54576	0%
Počet Slice LUT tabulkek	811	27288	2%
Počet Slice s nevyužitými Flip - Flop registry	687	822	83%
Počet Slice s nevyužitými LUT tabulkami	11	822	1%
Poččet plně využitých LUT-FF párů:	124	822	15%
Množství BRAM	124	116	82%
Množství BUFG/BUFGCTRL:	4	16	25%

Tab. 3.2: Tabulka využití hardwarových prostředků s použití register balancingu.

Název	Využito	Celkem	Poměr
Počet Slice Registrů	197	54576	1%
Počet Slice LUT tabulkek	564	27288	6%
Počet Slice s nevyužitými Flip - Flop registry	395	581	67%
Počet Slice s nevyužitými LUT tabulkami	17	581	2%
Poččet plně využitých LUT-FF párů:	169	581	29%
Množství BRAM	114	116	82%
Množství BUFG/BUFGCTRL:	4	16	25%

Timing

Bez využití register balancingu

- Minimální perioda 27.313ns
- Maximální frekvence: 36.613MHz

S register balancingem

- Minimální perioda 16.064ns
- Maximální frekvence: 62.252MHz

Zde si můžete povšimnout, že minimální perioda při použití register balancingu je téměř o dvojnásobek nižší. S register balancingem je tedy možné pracovní frekvenci tohoto modulu.

4 ZÁVĚR

Tato diplomová práce se věnuje návrhu a realizací modulu v jazyce VHDL pomocí FPGA. Tento modul byl navržen tak, aby dokázal číst obrazová data z paralelního rozhraní a ukládal do blokové paměti RAM. Dále aby četl obrazová data, převedl do sériového formátu a zaslal do LED řadiče.

V teoretické části byly popsány jednotlivé použité komponenty, způsob komunikace mezi komponenty, metody ukládání a čtení dat z blokové paměti, která byla řešená dvojitou snímkovou pamětí. Teoretická část se také věnuje implementaci gamma korekce, barevné hloubky a globálního jasu. A to za pomocí pulsně šířkové modulace nebo binární kódové modulace.

Praktická část se věnuje samotnému návrhu tohoto modulu. Je zde popsána komunikace mezi zdrojem dat a FPGA přes rozhraní PPI, řízení toku dat blokové paměti. Dále zde byly popsány simulace a reálné průběhy z osciloskopu na rozhraní mezi FPGA a ostatními komponenty. V závěru diplomové práce zde bylo popsáno množství využitých hardwarových prostředků bez a s použitím metod pro zvýšení pracovního kmitočtu.

V diplomové práci se podařilo navrhnout modul s požadovanými vlastnostmi tak, aby bylo možné jednoduše měnit jednotlivé parametry. Výšku, šířku s maximálním rozlišením 320x240, barevnou hloubku, gamma korekci a globální jas. Barevná hloubka a globální jas byly v konečné fázi řešené binární kódovou modulací. Zpočátku byla barevná hloubka implementována pulsní šířkovou modulací, ale poté se ukázalo, že je jí vhodnější implementovat binární kódovou modulací. Komunikace čtení dat z blokové paměti byla zprvu ukládána do dlouhých registrů. Bohužel to se ukázalo při velkém rozlišení jako slepá ulička. Poté tato část byla přepracována a čtená data ukládána do bufferu, která byla nutná pouze pro jeden hodinový impuls posuvného registru.

LITERATURA

[1] 32x16 and 32x32 RGB LED Matrix. Adafruit [online]. 2012 [cit. 2016-12-14]. Dostupné z: <https://learn.adafruit.com/32x16-32x32-rgb-led-matrix/overview>.

[2] Atlys FPGA Board Reference Manual [online]. Henley Court, 2016, 19 [cit. 2016-12-14]. Dostupné z: <https://reference.digilentinc.com/_media/atlys:atlys:atlys_rm.pdf>.

[3] Datasheet MBI5024. Macroblock [online]. Hsinchu, 2008, 18 [cit. 2016-12-14]. Dostupné z: <http://ledlamps.ru/MBI5024.pdf>.

[4] Gamma korekce. *Fotoroman.cz* [online]. Praha, 2011 [cit. 2017-05-23]. Dostupné z: <http://fotoroman.cz/glossary/3_gamma.htm>.

[5] HONSNEJMAN, Petr. *Barevná hloubka* [online]. 2016 [cit. 2017-05-23]. Dostupné z: <http://moje.tajemno.net/barevna-hloubka/>.

[6] KOLOUCH, Jaromír. Subsystémy používané v obvodech CPLD a FPGA a techniky jejich implementace [online]. [cit. 2016-12-14]. Dostupné z: <http://www.urel.feec.vutbr.cz/~kolouch/pld/data/DrStudTxt.pdf>.

[7] KUBÍČEK, Michal. Úvod do problematiky obvodů FPGA pro integrovanou výuku VUT a VŠB-TUO [online]. Brno: Vysoké učení technické v Brně, 2014 [cit. 2016-12-14]. ISBN 978-80-214-5069-1. Dostupné z: <https://vut-vsb.cz/home/get-file?file=427&%3Bportal=Portal2>.

[8] NEXPERIA. *Datasheet 74HC595; 74HCT595: 8-bit serial-in, serial or parallel-out shift register with output latches; 3-state* [online]. Nizozemí, 2016 [cit. 2017-05-23]. Dostupné z: <http://assets.nexperia.com/documents/data-sheet/74HC_HCT595.pdf>.

[9] PECH, Jan. Nebojte se FPGA. Vyvoj.hw.cz [online]. 2002, 10 [cit. 2016-12-14]. Dostupné z: <http://vyvoj.hw.cz/teorie-a-praxe/dokumentace/nebojte-se-fpga.html>.

[10] PPI: Parallel Peripheral Interface. Analog Devices [online]. 2011 [cit. 2016-12-14]. Dostupné z: <https://wiki.analog.com/resources/eval/sdp/sdp-b/peripherals/ppi>.

[11] Programovatelná logika II: FPGA [online]. ABC linuxu, 2013 [cit. 2016-12-14]. Dostupné z: <http://www.abclinuxu.cz/blog/digital_design/2013/1/programovatelna-logika-ii-fpga>.

[12] Spartan-6 FPGA Block RAM. XILINX [online]. 2011, v1.5, 34 [cit. 2016-12-14]. Dostupné z: <https://www.xilinx.com/support/documentation/user_guides/ug383.pdf>.

[13] Spartan-6 FPGA Data Sheet: DC and Switching Characteristics. Xilinx [online]. 2015, (v3.1.1), 89 [cit. 2016-12-14]. Dostupné z: <https://www.xilinx.com/support/documentation/data_sheets/ds162.pdf>.

[14] Using Binary Code Modulation to control LED brightness. *Hackaday.com* [online]. 2011 [cit. 2017-05-23]. Dostupné z: <http://hackaday.com/2011/07/22/using-binary-code-modulation-to-control-led-brightness/>.

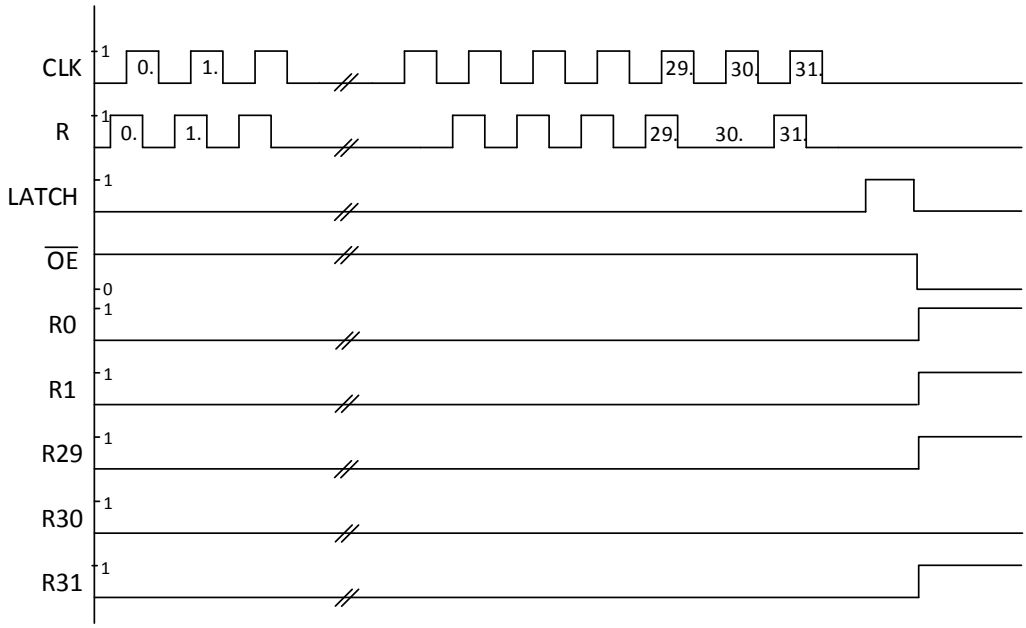
[15] Vmod MIB Reference Manual. Digilent [online]. Henley Court, 2011, 5 [cit. 2016-12-14]. <Dostupnéz:https://reference.digilentinc.com/_media/vmodmib/vmodmib_rm.pdf>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ASIC	Application-Specific Integrated Circuit – Aplikačně specifický integrovaný obvod
BCM	Binary code modulation – Binární kódová modulace
BRAM	Bloková RAM paměť
CLK	Clock – Hodinový signál
FPGA	Field-programmable gate array – Programovatelné hradlové pole
HDL	Hardware Design Language – Jazyk popisující návrh obvodu
HDMI	High-Definition Multi-media Interface – Digitální rozhraní audio a video signálu.
Hz	Hertz – Jednotka kmitočtu
I/O	Input/Output – Vstupně/výstupní
LED	Light Emiting Diode – Světlo emitující dioda
LUT	Look-Up Table
ms	milisekunda
ns	nanosekunda
PPI	Parallel Peripheral Interface – Paralelní periferní rohraní
PWM	Pulse Width Modulation – Pulzně šířková modulace
$\overline{OE}$	Output Enable – Výstup aktivován
RAM	Random-Access Memory – Paměť s libovolným výběrem
RGB	Red Green Blue – Červená zelená modrá
UCF	User Constraints File – Uživatelský nastavovací soubor
USB	Universal Serial Bus – Univerzální sériová sběrnice
VHDL	Very High Speed Integrated Circuits Hardware Description Language – Jazyk pro popis velmi rychlých integrovaných obvodů

SEZNAM PŘÍLOH

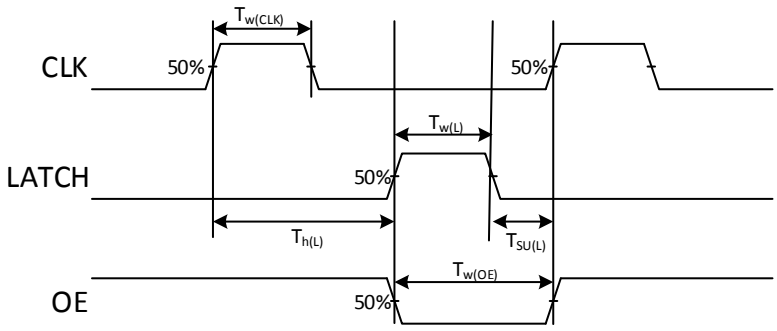
A INFORMACE O ŘADIČI MBI5024



Obr. A.1: Časové průběhy displeje Adafruit[3].

Tab. A.1: Tabulka minimálních časů signálů displeje [3].

Popis		Minimální hodnota
Šířka impulsu	CLK	20ns
	LATCH	20ns
	$\overline{OE}$	100ns
SETUP time pro LATCH		30ns
HOLD time pro LATCH		50ns



Obr. A.2: Časování mezi hodinovým signálem a ostatními signály [3].

B OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje naprogramovaný modul, včetně simulací a spouštěcího souboru. Také zde najdete diplomovou práci v elektronické podobě a vnitřní schéma LED displeje pro firmu Ing. Ivo Herman, CSc.

Modul je spustitelný v prostředí Xilinx ISE, verze 14.7, která je pro studenty volně ke stažení po zaregistrování na xilinx.com.

- `/` ..... kořenový adresář přiloženého CD
  - `xdolej04` ..... adresář praktické části
    - `sources` ..... adresář zdrojových kódů
      - `Top.vhdl` ..... hlavní soubor
      - `tb_Top.vhd` ..... simulace hlavního souboru
      - `detekce_hrany.vhd` ..... detekce hrany generatoru frekvence
      - `led_control.vhd` ..... ovládání displeje
      - `mytypes.vhd` ..... balík užitečných funkcí
      - `nacitani.vhd` ..... komunikace PPI rozhraní
    - `work` ..... pracovní adresář
      - `top.bit` ..... soubor pro nahrání do FPGA
    - `ipcore_dir` ..... ostatní
    - `isecofig`
    - `sp_6.ucf` ..... rozmístění pinů na FPGA
    - `xdolej04.xise` ..... spustitelný soubor projektu
  - `DL20x32x10B_40.pdf` ..... datasheet panelu 32x20
  - `xdolej04.pdf` ..... diplomová práce v elektronické podobě